***Make a document which includes all functions related to file operation with syntax and small example.***

**Basic file Operations :**

| | |
|---|---|
| fopen () | Creating a file or opening an existing file |
| fclose () | Closing a file |
| fprintf () | Writing a block of data to a file |
| fscanf () | Reading a block data from a file |
| getc () | Reads a single character from a file |
| putc () | Writes a single character to a file |
| fseek () | Sets the position of a file pointer to a specified location |
| ftell () | Returns the current position of a file pointer |

***To create a file in a 'C' program following syntax is used,***

```
FILE *fp;
fp = fopen ("file_name", "mode");
```

| File Mode | Description |
|---|---|
| r | Open a file for reading. If a file is in reading mode, then no data is deleted if a file is already present on a system. |
| w | Open a file for writing. If a file is in writing mode, then a new file is created if a file doesn't exist at all. If a file is already present on a system, then all the data inside the file is truncated, and it is opened for writing purposes. |
| a | Open a file in append mode. If a file is in append mode, then the file is opened. The content within the file doesn't change. |
| r+ | open for reading and writing from beginning |
| w+ | open for reading and writing, overwriting a file |
| a+ | open for reading and writing, appending to file |

***Example:***

```
#include <stdio.h>
int main() {
FILE *fp;
fp  = fopen ("data.txt", "w");
}
```

***You can specify the path where you want to create your file***

```
#include <stdio.h>
int main() {
FILE *fp;
fp  = fopen ("D://data.txt", "w");
}
```

## *fclose :*

```
fclose (file_pointer);
```

Example:

```
FILE *fp;
fp  = fopen ("data.txt", "r");
fclose (fp);
```

## *fputc() :*

```
#include <stdio.h>
int main() {
        int i;
        FILE * fptr;
        char fn[50];
        char str[] = "Guru99 Rocks\n";
        fptr = fopen("fputc_test.txt", "w"); // "w" defines "writing mode"
        for (i = 0; str[i] != '\n'; i++) {
            /* write to file using fputc() function */
            fputc(str[i], fptr);
        }
        fclose(fptr);
        return 0;
    }
```

## *fputs () :*

```
#include <stdio.h>
int main() {
        FILE * fp;
        fp = fopen("fputs_test.txt", "w+");
        fputs("This is Guru99 Tutorial on fputs,", fp);
        fputs("We don't need to use for loop\n", fp);
        fputs("Easier than fputc function\n", fp);
        fclose(fp);
        return (0);
    }
```

## *fprintf() :*

```
#include <stdio.h>
    int main() {
        FILE *fptr;
        fptr = fopen("fprintf_test.txt", "w"); // "w" defines "writing mode"
        /* write to file */
        fprintf(fptr, "Learning C with Guru99\n");
        fclose(fptr);
        return 0;
    }
```

## *fputs_test.txt , fgets() , fscan() , fgetc() :*

***The following program demonstrates reading from fputs_test.txt file using fgets(),fscanf() and fgetc () functions respectively :***

```
#include <stdio.h>
int main() {
        FILE * file_pointer;
        char buffer[30], c;

        file_pointer = fopen("fprintf_test.txt", "r");
        printf("----read a line----\n");
        fgets(buffer, 50, file_pointer);
        printf("%s\n", buffer);

        printf("----read and parse data----\n");
        file_pointer = fopen("fprintf_test.txt", "r"); //reset the pointer
        char str1[10], str2[2], str3[20], str4[2];
        fscanf(file_pointer, "%s %s %s %s", str1, str2, str3, str4);
        printf("Read String1 |%s|\n", str1);
        printf("Read String2 |%s|\n", str2);
        printf("Read String3 |%s|\n", str3);
        printf("Read String4 |%s|\n", str4);

        printf("----read the entire file----\n");

        file_pointer = fopen("fprintf_test.txt", "r"); //reset the pointer
        while ((c = getc(file_pointer)) != EOF) printf("%c", c);

        fclose(file_pointer);
        return 0;
    }
```

## *getc and putc:*

```
#include <stdio.h>
int main() {
        FILE * fp;
        char c;
        printf("File Handling\n");
        //open a file
        fp = fopen("demo.txt", "w");
        //writing operation
        while ((c = getchar()) != EOF) {
            putc(c, fp);
        }
        //close file
        fclose(fp);
        printf("Data Entered:\n");
        //reading
        fp = fopen("demo.txt", "r");
        while ((c = getc(fp)) != EOF) {
            printf("%c", c);
        }
        fclose(fp);
        return 0;
    }
```